Podrecznik uzytkownika programu

redagujaco-laczacego LINK

dla systemu CROOK

J. Milewski J. Rudzinski

IIUW Warszawa, marzec 1986

1. Informacje systemowe

W systemie binariow wystepuja zbiory modulow relokowalnych i absolutnych. Zbiory modulow relokowalnych moga zawierac jeden lub wiecej takich modulow, przy czym na zbior moze byc ewentualnie nalozona struktura biblioteki, wystepuja wiec zbiory sekwencyjne i biblioteczne.

Zbiory modulow o strukturze bibliotecznej roznia sie od zbiorow sekwencyjnych slownikiem dolaczonym na koncu zbioru, tzn. za ostatnim modulem wchodzacym w sklad biblioteki. Struktura zbiorow jest tak dobrana, ze we wszystkich kontekstach, w ktorych na wejsciu programu wymagany jest zbior sekwencyjny, moze rowniez wystapic biblioteka; istnieja natomiast sytuacje, w ktorych niezbedne jest istnienie na zbiorze struktury bibliotecznej, a zwykle zbiory sekwencyjne nie moga byc zastosowane.

Jezeli zbior o strukturze bibliotecznej przetwarzany jest w kontekscie, w ktorym dopuszczalny bylby tez zbior sekwencyjny, poszczegolne moduly biblioteki beda czytane przez program w nieokreslonej kolejnosci, zaleca sie wiec, aby programy laczone z wielu modulow zachowywaly sie niezmienniczo ze wzgledu na kolejnosc umieszczenia modulow w pamieci. W systemie istnieja metody, pozwalajace na precyzyjne lokowanie zadanych fragmentow kodu we wlasciwe miejsca pamieci.

Moduly relokowalne charakteryzuja sie nastepujacymi, widzialnymi przez programy je obslugujace atrybutami:

nazwa modulu,
adres transferowy: wyrozniony adres, pod ktory nalezy

przekazac sterowanie po zaladowaniu modulu; moze w module nie wystapic,
symbole eksportowane: (w skrocie eksporty) nazwy adresow wewnatrz modulu, ktore maja byc znane w innych modulach
symbole importowane: (w skrocie importy) nazwy adresow obiektow, ktore maja byc widziane w danym module, ale w nim nie wystepuja,
zmienne zewnetrzne: omowione beda pozniej.

Warto wiedziec, ze nazwy modulow nie maja nic wspolnego z nazwami obiektow zewnetrznych danego modulu, w szczegolnosci moga sie z takimi nazwami pokrywac bez zadnych konsekwencji dla sposbu wykorzystania modulu.

Kazdy eksport, import i zmienna zewnetrzna ma okreslony typ relokacji, a mianowicie Byte, Code lub Data, przy czym symbole z obszaru Byte wyrazone sa z bajtach, natomiast z obszarow Code i Data w slowach. Typz bajtach, natomiast z
obszarow Code i Data w slowach. Typ obszaru nie jest rozszerzeniem nazwy obiektu w tym sensie, ze obecnosc dwu obiektow o tej samej nazwie ale innym typie relokacji sygnalizowana jest jako blad.

2. Dzialanie LINKa

Link-editor LINK moze pracowac w dwoch trybach: relokowalnym i absolutnym. W trybie relokowalnym wynikiem pracy jest jeden modul relokowalny, ktory moze byc umieszczony w bibliotece lub laczony z innymi modulami w kolejnym przebiegu LINKa, nie moze natomiast byc wykonany.
W trybie absolutnym produkowany jest jeden zbior zawierajacy binaria absolutne, byc moze podzielone na rozne nakladki. Zbior ten moze byc zaladowany i wykonany, natomiast nie mozna go dalej przetwarzac tu opisanymi programami.
Zbiory wejsciowe do LINKa mozna podzielic na zbiory zwykle i biblioteki. Zbiory zwykle moga byc zbiorami sekwencyjnymi lub bibliotecznymi, natomiast biblioteki musza miec strukture biblioteczna. Ze zbiorow zwyklych trafiaja do modulu wynikowego wszystkie, badz wybrane explicite przez uzytkownika moduly, natomiast biblioteki przeszukiwane sa automatycznie i wybierane z nich sa te i tylko te moduly, ktore zawieraja eksporty, odpowiadajace nie zwiazanym jeszcze importom.
Omowimy najpierw zachowanie sie LINKa w trybie relokowalnym.

Adres transferowy moze wystapic w co najwyzej jednym module wejsciowyerowy moze wystapic w co najwyzej jednym
module wejsciowym i jesli taki wystepuje, przekazywany jest do modulu wynikowego.
Nazwa modulu wynikowego jest nazwa tego modulu wejsciowego, ktory zawieral adres transferowy, a jesli takiego nie ma, pierwszego napotkanego.
W komentarzu do modulu LINK podaje czas i date polaczenia

modulu i swoj identyfikator (LINK numer wersji), gubi natomiast ewentualne komentarze uzytkownika.
Nazwy importow sa uzgadniane z eksportami, przy czym nazwy nie uzgodnione przechodza do modulu wynikowego. Wsrod wszystkich modulow wejsciowych eksport o danej nazwie powinien wystapic tylko raz.
W roznych modulach wejsciowych moga wystapic zmienne zewnetrzne o tej samej nazwie, ktorym w module wynikowym odpowiadac bedzie jedna zmienna zewnetrzna o tej nazwie i o dlugosci najwiekszej z nich. Obiekt odpowiadajacy tej zmiennej powstanie poza cialem modulu wynikowego.
W wypadku wystapienia eksportu o tej samej nazwie co zmienna zewnetrzna, jest ona utozsamiana z tym eksportem (wchlaniana prze niego) i w module wynikowym juz nie wystepuje. Zmienna zewnetrzna o wyroznionej nazwie "$" nie moze zostac wchlonieta przez zaden eksport. Modul moze importowac nazwe zmiennej zewnetrznej.

Zarowno para eksport/import jak i zmienne zewnetrzne umozliwiaja dzielenie obszaru pamieci przez niezaleznie asemblowane moduly, jednak przy eksporcie wystepuje wyrazna asymetria: symbol eksportowany jest przez dokladnie jeden modul, w ktorym zarezerwowno odpowiedni obszar, a pozostale moduly ten symbol importuja, mamy wiec asymetrie sytuacji nadawcy i odbiorcy. Przy zmiennych zewnetrznych zachowana jest symetria: dowolna ilosc modulow moze deklarowac w identyczny sposob zmienna zewnetrzna o tej samej nazwie, ktora w rezultacie usytuowana bedzie przez system poza obszarem tych modulow. Przewiduje sie generowanie zmiennych zewnetrznych przez kompilator jezyka C (odpowiednik pojecia zmiennej zewnetrznej w tym jezyku) i przez kompilator fortranu (odpowiednik bloku COMMON z nazwa).

Przy pracy w trybie absolutnym LINKer wykonuje w sposob wyzej opisany sklejenie modulow tworzacych poszczegolne nakladki, a nastepnie ustala wzajemne polozenie nakladek, oblicza polozenie zmiennych zewnetrznych oraz generuje zbior zawierajacy binaria absolutne utworzonych nakladek. Wystepuje dodatkowy warunek, a mianowicie nakladka glowna - poziomu zerowego musi miec adres transferowy; pozostale nakladki taki adres miec moga.
Wewnatrz pliku wynikowego kazda nakladka winna miec unikalna nazwe, po ktorej rozpozna ja ekstrakod OVERLAY. Nazwa nakladki glownej tworzona jest w mysl tych samych regul, co nazwa modulu wynikowego przy pracy w trybie relokowalnym, natomiast nazwy pozostalych nakladek moga byc zadane dyrektywami LINKa, a przy ich braku ustalane sa rowniez w mysl tych samych regul.

Wszystkie nazwy importow i eksportow traktowane sa globalnie, tzn. w dowolnej nakladce mozna dysponowac adresem

symbolu eksportowanego z dowolnej innej nakladki, mozliwe jest wiec laczenie symboli "w dol" (procedura w nakladce korzysta z procedur swojego przodka), "w gore" (proces wywoluje procedure w zaladowanej nakladce) i "w bok" (adres obiektu z sasiedniej nakladki przekazywany jest w dol jako parametr).
W zwiazku z wyzej powiedzianym, dowolny modul przylaczony automatycznie z biblioteki pojawi sie co najwyzej raz. Jesli wiec w dwu roznych nakladkach ma byc wykorzystana ta sama procedura biblioteczna, nalezy ja dolaczyc we wspolnym przodku tych nakladek. Najwygodniej jest zapewne dolaczyc wszystkie procedury biblioteczne juz w nakladce poziomu zerowego.
W razie potrzeby mozna przygotowac nakladke wraz z modulem bibliotecznym osobnym przebiegiem LINKa w trybie relokowalnym, a nastepnie dolaczyc te nakladke z juz wbudowanym modulem bibliotecznym we wlasciwe miejsce struktury.

Istnieja pewne dodatkowe warunki na zmienne zewnetrzne w programach nakladkowanych. Zalozmy, ze zmienne zewnetrzne o tej samej nazwie name wystapily w nakladkach Ox1, Ox2,... Oxm. Niech D(name) = D(Ox1, Ox2,...Oxm) bedzie minimalnym poddrzewem zawierajacym te nakladki. Wtedy w korzeniu drzewa D(name) musi wystapic badz zmienna zewnetrzna o tej nazwie, badz wchlaniajacy ja eksport. Jesli w korzeniu wystepuje zmienna musi ona dodatkowo miec dlugosc niemniejsza, niz dlugosc wszystkich innych zmiennych zewnetrznych o tej nazwie.
Powyzsze ograniczenie nie dotyczy zmiennej zewnetrznej o wyroznionej nazwie "$", ktora moze wystapic w dowolnych nakladkach, natomiast nie moze byc wchlonieta przez eksport. Zmienna $ lokowana jest w wolnym obszarze pamieci, tak, aby mozna w niej bylo umiescic rozszerzalne struktury danych. W kompilatorze Fortranu zmienna ta bedzie odpowiadac blokowi COMMON bez nazwy.

Przy rozmieszczaniu binariow absolutnych w pamieci LINKer stosuje sie do nastepujacych zasad: Slowo o numerze zero jest zawsze wolne, nastepnie umieszczana jest zawartosc sekcji BYTE wszystkich nakladek, nastepnie zawartosc sekcji CODE i na koncu DATA. Oznacza to, ze sekcje B C i D nie beda nigdy przeplecione, natomiast przeplatanie moze dotyczyc sekcji roznych nakladek. Obszar wolny, nie zajety przez zadne sekcje, przeznaczony jest na zmienna zewnetrzna "$", ktora ma pelnic funkcje zmiennej rozszerzalnej. Mozliwe sa wiec nastepujace rozmieszczenia sekcji programu wynikowego, zaleznie od typu zmiennej zewnetrznej $:
[0] Byte Code Data (nie ma)
[0] Byte Code Data $-xvar [koniec pamieci] (typ data)
[0] Byte Code $-xvar Data [koniec pamieci] (typ code)
[0] Byte $-xvar Code Data [koniec pamieci] (typ byte)

Jesli program sklada sie z nakladek, wtedy LINK zostawia miejsce na poszczegolne sekcje danej nakladki miedzy sekcjami przodka. Jesli na przyklad nakladka poziomu jeden zawiera sekcje Code, wtedy w nakladce poziomu zero (korzeniu programu) poczatek sekcji Data bedzie tak usytuowany, aby caly obszar Code nakladki poziomu jeden zmiescil sie za obszarem Code a przed obszarem Data nakladki glownej.
Kazdy modul moze importowac symbole generowane przez LINKa, a opisujace poczatek kolejnych sekcji. Nazwy tych symboli to $.BYTE, $.CODE i $.DATA. Dysponujac tymi symbolami program moze obliczyc maksymalna dlugosc obszaru pamieci wolnego dla zmiennej zewnetrznej $.

3. Parametry wywolania programu LINK

Link-editor LINK uruchomiany jest przez podanie jego nazwy, po ktorej w linii wywolania mozna umiescic parametry, ktorych znaczenie opiszemy nizej:

```
LINK par1 par2 ...
```

W parametrach wywolania programu LINK mozna umiescic tytuly zbiorow i wskazania opcji. Tytuly zbiorow opisuja zbiory wejsciowe i biblioteczne. Istnieja nastepujce grupy opcji:

Opcje opisu bibliotek:

?LIB - wszystkie tytuly zbiorow wystepujace przed napotkaniem tej opcji opisuja zbiory wejsciowe LINKa, natomiast zbiory za ta opcja sa bibliotekami. Nawet w wypadku wzajemnych odwolan modulow w kilku bibliotekach kolejnosc podania tych bibliotek jest obojetna.
?NSL - (not system library) po przeszukaniu wskazanych przez uzytkownika bibliotek przegladana jest (o ile istnieje w systemie) biblioteka systejest (o ile
istnieje w systemie) biblioteka systemowa o nazwie /LIBRAR/SYSLIB. Ustawienie opcji ?NSL wstrzymuje przegladanie tej biblioteki.

Opcje trybu pracy:

?REL - ustawienie tej opcji powoduje generowanie przez LINKer modulu relokowalnego,
?ABS - ta opcja ustawia LINKa w tryb pracy z generowaniem modulu absolutnego.
Jako domyslny przyjety bedzie tryb pracy absolutny.

Opcje wykonywania komend:

?CMD - opcja ta powoduje czytanie przez LINKa komend ze

strumienia INT. Przy jej braku LINK automatycznie laczy moduly napotkane w zadanych zbiorach wejsciowych. Podanie wiecej niz jednego zbioru wejsciowego wyklucza uzycie opcji ?CMD.

?CMF tytul zbioru - jak wyzej, ale dyrektywy czytane beda z pliku, podanego jako nastepny parametr. Przy tym trybie pracy bledy syntaktyczne dyrektyw sa bledami fatalnymi.

Jesli przy wywolaniu nie bedzie podany ani jeden zbior wejsciowy, a takze nie wystapi zadna z opcji ?CMD lub ?CMF, jako zbior wejsciowy przyjety bedzie domyslnie STR//BI.

Opcje zbioru wyjsciowego:

?OUT tytul zbioru - wynik pracy LINKa wyslany bedzie na zbior dyskowy o podanej nazwie. Przy braku tego parametru LINK nie generuje zbioru wynikowego (przebieg testowy, generacja listingu).

Opcje zachowania eksportow:

?EXP - przy pracy w trybie relokowalnym do zbioru wynikowego trafiaja tylko te eksporty, ktorym nie odpowiadal zaden import. Po ustawieniu opcji ?EXP do modulu wynikowego przejda wszystkie eksporty. Opcja jest ignorowana przy pracy w trybie absolutnym.

Opcje listowania:

Domyslnie LINK wysyla jedynie komunikaty o bledach i ostrzezenia na strumien MON. Mozna ustawic jedna z trzech nastepujacych opcji listowania:

?NOL tytul zbioru - komunikaty o bledach i ostrzezenia beda dodatkowo wysylane do podanego zbioru.

?LIS tytul zbioru - do podanego zbioru bedzie wyslany opis wszystkich modulow wejsciowych (oprocz bibliotecznych) i opis modulu wynikowego badz nakladek wynikowych.

?FUL tytul zbioru - jak wyzej, a ponadto wygenerowana zostanie mapa referencji symboli zewnetrznych, podajaca, w ktorych modulach wystapilo odwolanie do danego symbolu. Dokladny opis bedzie generowany dla wszystkich modulow wejsciowych, w tym i bibliotecznych.

Tytuly zbiorow wystepujace po opcjach ?OUT, ?CMF i listowania nie sa traktowane jako opisy zbiorow wejsciowych ani bibliotek.

4. Dyrektywy LINKa

W trybie pracy z dyrektywami mozliwe sa nastepujace dyrektywy:

L['tytul zbioru] [nazwa modulu]*

ta dyrektywa oznacza dolaczenie modulow o wskazanych nazwach (przy ich braku wszystkich modulow) z podanego zbioru. Przy braku nazwy zbioru przyjmuje sie zbior uzyty w poprzedniej dyrektywie L lub, dla pierwszej takiej dyrektywy, zbior podany jako parametr wejsciowy.

On [nazwa nakladki]

Ta dyrektywa dopuszczalna jest jedynie w trybie absolutnym i pozwala na kreowanie drzewa nakladek. Liczba n z przedzialu
n
n
1..9 oznaczajaca poziom nakladki moze byc wieksza o jeden od numeru nakladki biezacej przy tworzeniu kolejnych, wyzszych nakladek, badz mniejsza/rowna od numeru biezacego przy tworzeniu nowych galezi bocznych. Nazwe nakladki mozna opuscic, przyjeta bedzie wtedy nazwa modulu z adresem transferowym, a w jego braku nazwa pierwszego napotkanego modulu.

LINK w fazie inicjalizacji wykonuje wewnetrzna dyrektywe

O0

natomiast w trakcie pracy tej dyrektywy nie mozna powtorzyc: moze istniec dokladnie jedna nakladka poziomu zero.

Q

Dyrektywa Quit inicjuje tworzenie zbioru wynikowego, a po jego utworzeniu konczy prace.

Q!

Ta dyrektywa konczy prace bez tworzenia zbioru wynikowego.

Poczet liter w nazwach dyrektyw i tytulach zbiorow jest nieistotny, natomiast w nazwach modulow i nakladek male i duze litery sa rozrozniane.

Separatorem elementow dyrektyw moze byc ciag spacji, badz przecinek, ewentualnie otoczony spacjami.

5. Ladowanie binariow absolutnych

Jesli program nie zawiera nakladek, to zaladowaniem i

wykonaniem programu zajmie sie system operacyjny, gdyz plik z binariami absolutnymi utworzony przez LINKa w stosownym trybie pracy jest zbiorem ladowalnym, uruchomianym przez jezyk sterowania praca.
Podanie nazwy takiego pliku spowoduje zaladowanie programu w przydzielony obszar pamieci i uruchomienie go od adresu transferowego. Program bedzie dysponowal parametrami wywolania w sposob zgodny z konwencjami systemu operacyjnego.

Program podzielony na nakladki ladowany jest wstepnie tak, jak program bez nakladek, z tym, ze system umieszcza w pamieci i uruchamia od adresu transferowego jedynie nakladke poziomu zero. Aby umozliwic ladowanie pozostalych nakladek, program musi byc zaopatrzony w umieszczone explicite przez programiste zadania zaladowania wskazanej nakladki. Sluzy do tego ekstrakod OVERLAY, ktorego parametrem jest nazwa nakladki i opcjonalnie adres, pod ktory nalezy przekazac sterowanie po wyjsciu z ekstrakodu.

Efektem pozytywnego wykonania ekstrakodu OVERLAY jest umieszczenie w pamieci operacyjnej wartosci podanych w tabliczkach opisujacych nakladke o zadanej nazwie, przy czym zawartosc zadnych innych slow pamieci nie ulegnie zmianie. Jesli nakladka zawierala adres transferowy, sterowanie po wyjsciu z ekstrakodu przejdzie pod ten adres, a w przeciwnym wypadku wroci w miejsce wywolania ekstrakodu. Jesli bedzie podany adres powrotu jako parametr ekstrakodu, sterowanie bedzie tam wlasnie przekazane, a ewentualnie istniejacy adres transferowy zostanie zignorowany.

Dodatek A. Struktura zbiorow binarnych jako plikow

Zbiory binariow relokowalnych sekwencyjne i biblioteczne zawieraja ciag umieszczonych jeden za drugim modulow. W zbiorach sekwencyjnych za ostatnim modulem wystepuje koniec pliku, natomiast w zbiorach bibliotecznych umieszcza sie tam kartoteke.

W systemie CROOK zbiory binariow relokowalnych maja typ OBJ, a zbiory binariow absolutnych sa typu BIN. Dla odroznionych binariow absolutnym od zbiorow binarnych dawnego typu uzywany jest bit 4 parametru ATR, ktory bedzie ustawiany wstepneie na O'64').
Parametr LEN wyzej wymienionych typow zbiorow zawiera jak zwykle dlugosc zbioru mierzona w blokach, parametr PAR1 = LEN -1, a parametr PAR2 zawiera offset ostatniego slowa w ostatnim bloku, tzn. PAR2 = 0 oznacza, ze istotne jest jedno slowo ostatniego bloku, a PAR2 = 255 znaczy, ze ostatni blok jest pelny.